

- 1 -

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	:	Before the Examiner:
Goodman et al.	:	Nalven, Andrew L.
Serial No.: 09/931,550	:	Group Art Unit: 2434
Filing Date: August 16, 2001	:	
	:	
Title: SYSTEM MANAGEMENT	:	Lenovo (United States) Inc.
INTERRUPT GENERATION UPON	:	Building 675, Mail C-137
COMPLETION OF CRYPTOGRAPHIC	:	4401 Silicon Drive
OPERATION	:	Durham, NC 27709
	:	

SUPPLEMENTAL APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. **REAL PARTY IN INTEREST**

Lenovo (Singapore) Pte. Ltd. is the assignee of the entire right, title and interest in the above-identified patent application.

II. **RELATED APPEALS AND INTERFERENCES**

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. **STATUS OF CLAIMS**

Claims 3-9 and 12-19 are pending in the Application. Claims 1-2 and 10-11 were cancelled. Claims 3-9 and 12-18 stand rejected. Claim 19 is allowed.

IV. **STATUS OF AMENDMENTS**

Appellants have not submitted any amendments following receipt of the final rejection with a mailing date of July 28, 2005.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 4:

In a data processing system, a method for updating a utility, comprising the steps of: (Specification, page 8, lines 17-18; Specification, page 9, lines 1-2; Figure 4, element 413)

receiving a request to unlock the utility; (Specification, page 9, lines 4-6; Figure 1, step 101)

verifying an update to the utility; (Specification, page 9, lines 23-24; Figure 2, step 202)

using a system management interrupt (SMI) handler to query a status of the verifying step; (Specification, page 9, lines 6-8 and 23-26)

and

if the verifying step successfully verifies the update of the utility, unlocking the utility and updating the utility, wherein the verifying step is performed by a trusted platform module (TPM) in accordance with Trusted Computing Platform Alliance Specifications. (Specification, page 8, lines 20-22; Specification, page 11, lines 4-8; Figure 3, steps 302, 303; Figure 4, element 451)

Independent Claim 13:

A computer program product for storage on a computer readable medium and operable for updating a utility, comprising: (Specification, page 7, line 20 – page 8, line 4)

programming for receiving a request to unlock the utility; (Specification, page 9, lines 4-6; Figure 1, step 101)

programming for verifying an update to the utility; (Specification, page 9, lines 23-24; Figure 2, step 202)

programming for using a system management interrupt (SMI) handler to

query a status of the verifying programming; and (Specification, page 9, lines 6-8 and 23-26)

if the verifying programming successfully verifies the update of the utility, programming for unlocking the utility and updating the utility, wherein the verifying programming is performed by a trusted platform module (TPM) in accordance with Trusted Computing Platform Alliance Specifications. (Specification, page 8, lines 20-22; Specification, page 11, lines 4-8; Figure 3, steps 302, 303; Figure 4, element 451)

Independent Claim 18:

A data processing system comprising: (Specification, page 7, lines 4-5; Figure 4, element 413)

a processor; (Specification, page 7, lines 5-6; Figure 4, element 410)

a trusted platform module (TPM) coupled to the processor and operating under Trusted Computing Platform Alliance Specifications; (Specification, page 8, lines 20-22; Figure 4, elements 410, 451)

a BIOS utility stored in flash memory coupled to the processor; (Specification, page 7, lines 6-8; Specification, page 8, lines 17-18; Specification, page 9, lines 2-3; Figure 4, elements 410, 416)

an input circuit for receiving an update to the BIOS utility; (Specification, page 9, lines 2-4) and

a bus system for coupling the input circuit to the processor; (Specification, page 7, lines 5-6; Figure 4, elements 410, 412)

a BIOS update application requesting an unlock of the flash memory from a system management interrupt (SMI) handler; (Specification, page 9, lines 4-6; Figure 1, step 101)

the SMI handler including programming for requesting cryptographic verification of the BIOS utility update from the TPM; (Specification, page 9, lines 24-26; Figure 2, step 203; Figure 4, element 451)

the TPM including programming for verifying an authenticity of the BIOS utility update; (Specification, page 10, lines 15-16)

the TPM including programming for issuing an SMI to query the TPM for a status on the verifying of the authenticity of the BIOS utility update; (Specification, page 10, lines 7-9; Figure 3, step 301)

the SMI handler unlocking the flash memory if the SMI handler sets the status as successful; (Specification, page 11, lines 12-15)

the BIOS update application updating the BIOS utility with the update (Specification, page 11, lines 17-19; Figure 1, step 105); and

the SMI handler locking the flash memory after the update of the BIOS utility has completed. (Specification, page 11, lines 17-19)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 3-9 and 12-18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over *Alexander et al.* (U.S. patent No. 6,188,602) in view of *Grawrock* (U.S. Patent No. 6,678,833).

VII. ARGUMENT

Claims 4-6, 13-15 and 18 stand rejected under 35 U.S.C. § 103 as being unpatentable over *Alexander* in view of *Grawrock* (U.S. Patent No. 6,678,833). In response, Applicants respectfully traverse these rejections.

The Examiner is attempting to combine *Grawrock* and *Alexander* in an impermissible manner. Nothing within *Alexander* teaches or suggests a need or even a hint for using a TPM such as taught in *Grawrock*.

The Examiner asserts that the motivation to combine the two references is provided in *Grawrock* at column 2, lines 1-6. *Grawrock* teaches that the TPM is bound physically or logically to the boot block memory device, such as shown in Figure 2 of *Grawrock*. This resulting configuration allows the TPM to accurately report the identity of the boot block without reliance on any intervening devices. Though a combination of *Grawrock* and *Alexander* may suggest that a TPM can be used to verify the identity of a boot block code, it does not suggest an ability to use a TPM to verify and update to such a boot block code, or especially a BIOS utility, such

as recited in some of the claims. In fact, *Grawrock* teaches away from the present invention by specifically stating that it uses the TPM so that there is no reliance on any intervening devices, while the present invention uses such an intervening device through the utilization of the SMI handler to query a status of the verifying step. In other words, *Grawrock* has the TPM so physically or logically connected to the boot block memory unit that it does not require such utilities as an SMI handler to assist it in verifying updates that may be desired to be stored on such a memory unit. The Examiner responds that an SMI handler is not a "device." It appears the Examiner is taking an overly narrow interpretation of the term "device," contrary to normal PTO practice.

With respect to claims 5 and 14 the Examiner asserts that the combination of *Alexander* and *Grawrock* would teach that an SMI handler could be used to query the status of the verifying step by querying the TPM for such status. The Examiner cites *Grawrock*, column 4, lines 1-9. This language in *Grawrock*, however, teaches that the TPM can be used to perform a hash operation on various software modules to produce an identifier that is then stored within the TPM, and then can be used to later respond to challengers wanting to verify the authenticity of such software modules. Column 3, line 50 - column 4, line 18. What is important is that the combination of these two references does not teach or suggest that an update to one of these software modules, and specifically the BIOS (as recited in several of the claims), is performed by the TPM before an update of such software module is accomplished. *Grawrock* teaches verification after it has already been loaded onto the system, whereas the present invention teaches a way to verify the BIOS is unaltered before allowing it to be flashed onto the system. This is the same difference as between catching the criminal after the crime has been committed versus preventing the crime.

Column 5, lines 41-45, does not teach "if the verifying step successfully verifies the update of the utility, unlocking the utility and updating the utility." The "unlocking" step is missing in *Alexander's* disclosure. And, yes, the claims do recite that verification must be completed. That is what "successfully verifies" means.

With respect to claims 6 and 15, the Examiner has asserted that the combination of *Alexander* and *Grawrock* teaches the SMI handler being issued by the TPM. This is in no way suggested by these two combinations. The Examiner cannot make such an assertion without attempting to at least prove it with some logical reasoning. The Examiner's assertion on page 5 of the Office Action is merely an unsupported single-sentenced statement.

With respect to claim 18, the foregoing arguments also apply.

VIII. CONCLUSION

For the reasons noted above, the rejections of claims 3-9 and 12-18 are in error. Appellants respectfully request reversal of the rejections and allowance of claims 3-9 and 12-19.

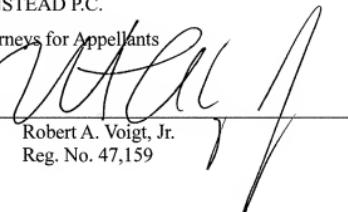
Respectfully submitted,

WINSTEAD P.C.

Attorneys for Appellants

By:

Robert A. Voigt, Jr.
Reg. No. 47,159



P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

CLAIMS APPENDIX

3. The method as recited in claim 4, further comprising the step of:
not unlocking the utility if the verifying step fails to verify the update to the utility.
4. In a data processing system, a method for updating a utility, comprising the steps of:
receiving a request to unlock the utility;
verifying an update to the utility;
using a system management interrupt (SMI) handler to query a status of the verifying step;
and
if the verifying step successfully verifies the update of the utility, unlocking the utility and updating the utility, wherein the verifying step is performed by a trusted platform module (TPM) in accordance with Trusted Computing Platform Alliance Specifications.
5. The method as recited in claim 4, wherein the SMI handler used to query the status of the verifying step queries the TPM for the status.
6. The method as recited in claim 5, wherein the SMI handler is issued by the TPM.
7. The method as recited in claim 4, further comprising the step of:
after the utility has been updated, locking the utility with the SMI handler.
8. The method as recited in claim 4, wherein the utility is a flash utility.
9. The method as recited in claim 4, wherein the requesting step is performed by an SMI handler.
12. The computer program product as recited in claim 13, further comprising:
programming for not unlocking the utility if the verifying programming fails to verify the update to the utility.

13. A computer program product for storage on a computer readable medium and operable for updating a utility, comprising:

programming for receiving a request to unlock the utility;

programming for verifying an update to the utility;

programming for using a system management interrupt (SMI) handler to query a status of the verifying programming; and

if the verifying programming successfully verifies the update of the utility, programming for unlocking the utility and updating the utility, wherein the verifying programming is performed by a trusted platform module (TPM) in accordance with Trusted Computing Platform Alliance Specifications.

14. The computer program product as recited in claim 13, wherein the SMI handler used to query the status of the verifying programming queries the TPM for the status.

15. The computer program product as recited in claim 14, wherein the SMI handler is issued by the TPM.

16. The computer program product as recited in claim 13, further comprising:

after the utility has been updated, programming for locking the utility with the SMI handler.

17. The computer program product as recited in claim 13, wherein the requesting programming is performed by an SMI handler.

18. A data processing system comprising:

a processor;

a trusted platform module (TPM) coupled to the processor and operating under Trusted Computing Platform Alliance Specifications;

a BIOS utility stored in flash memory coupled to the processor;

an input circuit for receiving an update to the BIOS utility; and

a bus system for coupling the input circuit to the processor;

a BIOS update application requesting an unlock of the flash memory from a system management interrupt (SMI) handler;

the SMI handler including programming for requesting cryptographic verification of the BIOS utility update from the TPM;

the TPM including programming for verifying an authenticity of the BIOS utility update;

the TPM including programming for issuing an SMI to query the TPM for a status on the verifying of the authenticity of the BIOS utility update;

the SMI handler unlocking the flash memory if the SMI handler sets the status as successful;

the BIOS update application updating the BIOS utility with the update; and

the SMI handler locking the flash memory after the update of the BIOS utility has completed.

19. A method comprising the steps of:

(a) a BIOS update application requesting an unlock of a flash utility from a system management interrupt (SMI) handler;

(b) determining if a verification of an update to the flash utility is pending;

(c) if verification of the update to the flash utility is not pending, the SMI handler requesting verification of the update to the flash utility from a trusted platform module (TPM) and setting a status flag as pending;

(d) exiting the SMI handler and returning status flag to the BIOS update application;

(e) receiving by the BIOS update application the status flag from the SMI handler;

(f) returning to step (a) if the status flag is set as pending after step (e);

(g) in response to step (c), the TPM verifies the update to the flash utility;

(h) when step (g) is completed, issuing an SMI by the TPM to query if the verification of the update to the flash utility was successful or failed;

(i) setting the status flag as successful if the verification of the update to the

flash utility was successful;

(j) setting the status flag as failed if the verification of the update to the flash utility was not successful;

(k) if step (b) determines that verification of the update to the flash utility is still pending, determining if the verification of the update to the flash utility has completed;

(l) if step (k) determines that verification of the update to the flash utility has not completed, setting the status flag as pending;

(m) if step (k) determines that verification of the update to the flash utility has completed, determining if the verification of the update to the flash utility was successful;

(n) if step (m) determines that the verification of the update to the flash utility was not successful, setting the status flag as failed;

(o) if step (m) determines that the verification of the update to the flash utility was successful, the SMI handler unlocking the flash utility and setting the status flag as successful;

(p) performing steps (d) and (e) in response to any of steps (l), (n), or (o);

(q) determining if the status flag is set as successful if after step (e) it is determined that the status flag is not set to pending; and

(r) updating the BIOS with the update to the flash utility and locking the flash utility with the SMI handler if the status flag is determined to be set to successful in step (q).

EVIDENCE APPENDIX

No evidence was submitted pursuant to §§1.130, 1.131, or 1.132 of 37 C.F.R. or of any other evidence entered by the Examiner and relied upon by Appellants in the Appeal.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings to the current proceeding.